Title: Requantization Transcoding for H.264/AVC Video Coding

Authors: Jan De Cock, Stijn Notebaert, Peter Lambert, and Rik Van de Walle

In: Signal Processing: Image Communication, Vol. 25 (4), pp. 235-254, April 2010.

Link: http://dx.doi.org/10.1016/j.image.2010.01.006

Requantization Transcoding for H.264/AVC Video Coding

Jan De Cock, Stijn Notebaert, Peter Lambert,

Rik Van de Walle

Ghent University – IBBT

Department of Electronics and Information Systems – Multimedia Lab

Gaston Crommenlaan 8 bus 201, B-9050 Ledeberg-Ghent, Belgium

Abstract

In this paper, efficient solutions for requantization transcoding in H.264/AVC are presented. By requantizing residual coefficients in the bitstream, different error components can appear in the transcoded video stream. Firstly, a requantization error is present due to successive quantization in encoder and transcoder. In addition to the requantization error, the loss of information caused by coarser quantization will propagate due to dependencies in the bitstream. Because of the use of intra prediction and motion-compensated prediction in H.264/AVC, both spatial and temporal drift propagation arise in transcoded H.264/AVC video streams. The spatial drift in intra-predicted blocks results from mismatches in the surrounding prediction pixels as a consequence of requantization. In this paper, both spatial and temporal drift components are analyzed. As is shown, spatial drift has a determining impact on the visual quality of transcoded video streams in H.264/AVC. In particular, this type of drift results in serious distortion and disturbing artifacts in the transcoded video stream. In order to avoid the spatially propagating

distortion, we introduce transcoding architectures based on spatial compensation techniques. By combining the individual temporal and spatial compensation approaches and applying different techniques based on the picture and/or macroblock type, overall architectures are obtained that provide a trade-off between computational complexity and rate-distortion performance. The complexity of the presented architectures is significantly reduced when compared to cascaded decoder-encoder solutions, which are typically used for H.264/AVC transcoding. The reduction in complexity is particularly large for the solution which uses spatial compensation only. When compared to traditional solutions without spatial compensation, both visual and objective quality results are highly improved.

Key words: transcoding, requantization, H.264/AVC, drift propagation, drift compensation.

---

1 Introduction

Video adaptation is required in order to meet network and user constraints in the multimedia content delivery chain. In many cases, reduction of the bit rate of video bitstreams is necessary. Requantization transcoding is a fast and elegant solution for reducing the bit rate, by decreasing the amount of residual data in the bitstream.

For MPEG-1/2 bitstreams, different solutions for transcoding have been investigated. In MPEG-1/2, variable length entropy codes allowed the use of fast coefficient clipping techniques, such as constrained or unconstrained dynamic rate shaping (DRS) [1,2]. Apart from DRS techniques, the major part of the investigated architectures focused on requantization transcoding. In [3], a single-loop transcoder system was derived starting from a cascaded decoder-

encoder solution, by merging common operations in the decoder and encoder loop. In [4], a number of reduced-complexity architectures, including an open-loop system, were investigated. Open-loop requantization transcoding proved to be a viable solution for MPEG-1/2, with a performance very similar to a decoder-encoder cascade [5]. Open-loop requantization transcoding is also the technique which is used in various commercial MPEG-1/2 rate shaping systems. Drift from open-loop requantization, however, may result in degraded performance when using longer GOP structures, due to temporal proliferation of requantization errors. In order to overcome this issue, drift-free solutions have been developed for MPEG-2 [6]. Owing to transform-domain operations, the complexity of these solutions was kept significantly lower than cascaded decoder-encoder architectures. In particular, solutions were sought for simplifications of the motion-compensated prediction (MCP) process during transcoding, resulting in efficient techniques for transform-domain MCP (MC-DCT) [6].

The H.264/AVC specification introduces advanced coding techniques [7] in order to increase the coding efficiency compared with previous video coding standards, e.g., MPEG-1/2 Video, H.263, and MPEG-4 Visual. This also results in more dependencies in the coded bitstream, which have to be taken into account during transcoding. Because of this, previously existing transcoding techniques are rendered obsolete. As an example, open-loop requantization can be applied to H.264/AVC bitstreams, but the visual quality of the transcoded bitstreams will be substantially affected. In particular, drift will propagate, both spatially and temporally. Hence, updated, intelligent techniques are required in order to reduce the bit rate.

Most of the research in literature on transcoding for H.264/AVC has been

dedicated to cascaded decoder-encoder solutions. Different fast motion estimation algorithms have been proposed. Mode refinement for a cascaded decoder-encoder in requantization transcoding was for example discussed in [8,9]. Solutions for spatial resolution reduction [10–12] or heterogeneous transcoding from or to H.264/AVC [13,14] also focused on pixel-domain cascaded decoder-encoder solutions. In particular the problem of fast motion estimation and mode decision using motion information from the incoming bitstream was studied.

In [15], an assessment was made of the performance of various existing requantization techniques, including a cascaded pixel-domain transcoder (CPDT) and a single-loop transcoder with temporal compensation (fast pixel-domain transcoder, FPDT). Another mixed architecture (MRA) was examined, which uses the CPDT architecture for intra-coded pictures, and FPDT for predictive (P) and bidirectionally predicted (B) pictures. Although performance was significantly increased, MRA led to unpredictable results, and rapidly degrading quality in GOP structures. A gap of about 3 to 4 dB in rate-distortion performance was found when compared to the CPDT solution. In this context, the problem of requantization of intra-coded regions in P and B pictures was remarked, but not resolved.

In this paper, we discuss requantization transcoding for H.264/AVC. Results show that open-loop requantization transcoding introduces severe quality degradation and therefore can not be used as bit rate reduction technique for intra-coded pictures. This is to a large extent caused by spatial drift due to intra prediction. We start with an analysis of the drift, by comparing the imperfect open-loop requantization transcoder and the drift-free cascaded pixel-domain architecture. We show that intra prediction is the major source of drift

in H.264/AVC transcoding, leading to disturbing artifacts in the transcoded video streams.

The organization of the paper is as follows. The paper starts with an analysis of drift errors in open-loop transcoding in Sect. 2. A brief overview of quantization in H.264/AVC is given, followed by a description of the quantizer changes which are required for proper requantization transcoding. Next, we examine spatial and temporal drift in H.264/AVC, caused by intra prediction and MCP, respectively. In order to identify both spatial and temporal drift components, the open-loop transcoder is compared to the drift-free CPDT approach. In Sect. 3, transcoding architectures with spatial compensation are discussed which reduce requantization error drift. In Sect. 3.4, combined architectures are presented, which apply spatial and/or temporal compensation based on the picture and macroblock type. In the results section, it is shown that architectures with spatial compensation highly improve the objective and visual quality of the output sequences when compared to the open-loop and traditional [1] FPDT architectures without spatial compensation. Conclusions are given in Sect. 5.

[1] By 'traditional architectures', we refer to the open-loop and single-loop architectures, as used in previous video coding standards, such as MPEG-2 and MPEG-4 Visual.

2 Requantization for H.264/AVC

2.1 Quantization in H.264/AVC

The brief overview of the H.264/AVC specification in this section provides a background for the open-loop requantization problem elaborated on in the next paragraphs. More details regarding the transform and quantization in H.264/AVC can be found in [16].

The quantization design in H.264/AVC is different when compared to quantization schemes of previous video coding specifications, due to the fact that floating point arithmetic is avoided (16-bit integer operations are sufficient for 8-bit pixel data), the quantization step size increases exponentially in order to provide much broader range of possible bit rates, and pre- and post-scaling operations of the forward and inverse integer transforms are incorporated in the quantization multiplier coefficients.

The H.264/AVC specification defines a scalar quantizer with 52 predefined quantization step (Q_{step}) sizes, corresponding to a quantization parameter (QP) ranging from 0 to 51. The quantization step size is doubled for an increase of the quantization parameter by 6, roughly corresponding to a decrease of the bit rate by half.

The forward and inverse transform and quantization processes are illustrated in Fig. 1, where the subscript $_{ij}$ indicates the position in the 4×4 matrix.

At the encoder side, the forward transform is denoted as $(\boldsymbol{C}_F\boldsymbol{X}\boldsymbol{C}_F^T) \otimes \boldsymbol{E}_F$ where $\boldsymbol{X}$ represents the input matrix, $\boldsymbol{C}_F$ the kernel forward transform ma-

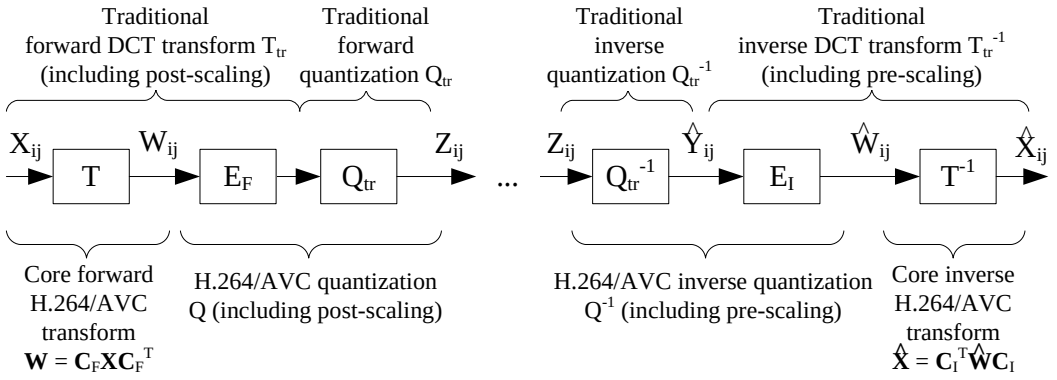


Fig. 1. Transform, scaling, and quantization in H.264/AVC.

trix and $\boldsymbol{E}_F$ the post-scaling matrix for normalization ($\otimes$ denotes Hadamard matrix multiplication). In order to reduce the computational complexity of transform and quantization, the post-scaling operation $\boldsymbol{E}_F$ is postponed to the quantization process, so the *core* forward transform in H.264/AVC can be described as $\boldsymbol{W} = \boldsymbol{C}_F\boldsymbol{X}\boldsymbol{C}_F^T$. After the core forward transform is applied, the quantization can be performed, which incorporates the post-scaling operation $\boldsymbol{E}_F$. Note that the H.264/AVC standard does not specify the forward quantization process. Instead, we can regard the following formula as a preferred way of performing quantization, given the reconstruction formulas in the standard:

$$|Z_{ij}| = (|W_{ij}| \cdot M_{ij} + \epsilon \cdot 2^{qbits}) \gg qbits \tag{1}$$

with $\text{sign}(Z_{ij}) = \text{sign}(W_{ij})$. Here, the parameter ϵ controls the dead zone size of the quantizer characteristic, and $\gg$ denotes the right shift operation. The H.264/AVC Joint Model reference software uses values of $\epsilon = 1/3$ for intra coding and $\epsilon = 1/6$ for inter coding. The parameters $qbits = 15 + \lfloor QP/6 \rfloor$ and M_{ij} both define the coarseness of the quantizer characteristic. M_{ij} is defined as:

$$M_{ij} = \text{round}\Big((2^{qbits} \cdot E_{F,ij})/Q_{step}\Big) \,. \tag{2}$$

As mentioned, the post-scaling factors $E_{F,ij}$ represent the normalization values

for the forward DCT transform. Since the odd and even base functions of the core forward transform have different norms, these values become matrix position-dependent. By including these values in the quantization, different quantizer multiplier coefficients are obtained depending on the position in the 4×4 block of transform coefficients. Hence, the forward multiplier factors M_{ij} depend not only on the quantization parameter, but they also become position dependent in the 4×4 block according to the following partitioning (corresponding to a combination of two even, two odd, or an even and an odd base function in the transform, respectively):

$$r = \begin{cases} 0, & (i,j) \in \{(0,0),(0,2),(2,0),(2,2)\} \\ 1, & (i,j) \in \{(1,1),(1,3),(3,1),(3,3)\} \\ 2, & \text{otherwise} \end{cases}$$

where (i, j) indicates the position in the 4×4 block of transform coefficients.

At decoder side, the inverse quantization reconstructs the transform coefficients $\widehat{W}_{ij}$ using the quantized values Z_{ij}:

$$\widehat{W}_{ij} = Z_{ij} \cdot V_{ij} \cdot 2^{\lfloor QP/6 \rfloor} \tag{3}$$

where the inverse multiplier coefficients V_{ij} are derived as follows:

$$V_{ij} = \text{round}\Big(\frac{64 \cdot Q_{step} \cdot E_{I,ij}}{2^{\lfloor QP/6 \rfloor}}\Big) \, . \tag{4}$$

As can be seen from the definition, the inverse quantization incorporates the pre-scaling operation with matrix $\boldsymbol{E}_I$ of the inverse transform $\boldsymbol{C}_I^T(\widehat{\boldsymbol{Y}} \otimes \boldsymbol{E}_I)\boldsymbol{C}_I$, so the *core* inverse transform in H.264/AVC is described as $\widehat{\boldsymbol{X}} = \boldsymbol{C}_I^T \widehat{\boldsymbol{W}} \boldsymbol{C}_I$. The scaling factors $E_{I,ij}$ are derived as normalization values for the inverse transform, hence these factors are matrix position-dependent.

In the remainder of the document, we denote the *core* forward and inverse transforms as T and T^{-1}, respectively, i.e., $T(\boldsymbol{X}) = \boldsymbol{C}_F \boldsymbol{X} \boldsymbol{C}_F^T$ and $T^{-1}(\widehat{\boldsymbol{W}}) = \boldsymbol{C}_I^T \widehat{\boldsymbol{W}} \boldsymbol{C}_I$. By Q and Q^{-1}, we indicate the forward and inverse H.264/AVC quantization processes, including post- and pre-scaling normalization operations.

2.2 Open-loop requantization for H.264/AVC

In this paragraph, we examine H.264/AVC requantization for the most straightforward requantization system, i.e., open-loop transcoding, and derive position-independent multiplier coefficients for requantization. In traditional open-loop systems, the transcoder is a simple concatenation of an inverse and forward quantization step. This system is shown in Fig. 2. Other syntax elements (macroblock partitioning, mode decision, motion vectors, etc.) are bypassed to the output bitstream.

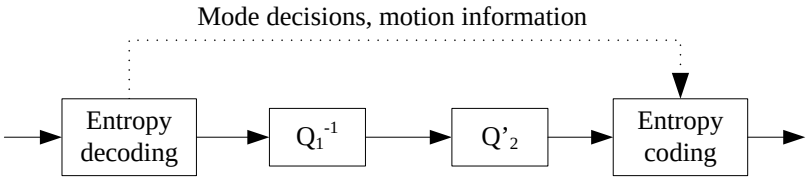


Fig. 2. Open-loop transcoder.

2.2.1 Derivation of multiplier coefficients for requantization

For H.264/AVC requantization, a number of elements in the design further complicate this system. As noted in the previous section, the intertwined transform and quantization lead to the incorporation of normalization values in the quantization, as was illustrated in Fig. 1.

Because these normalization values are already included in the encoder and decoder, respectively, care has to be taken not to repeat the scaling operation during transcoding. If not, the requantization process would produce values which are upscaled by a factor $64 \cdot E_{F,ij} \cdot E_{I,ij}$, which equals 4 ($r = 0$), 3.2 ($r = 1$), or 2.56 ($r = 2$). Hence, we propose to eliminate the factors $E_{F,ij}$ and $E_{I,ij}$ from the formulas above, resulting in position-independent requantization multiplier coefficients M' and V' (only the scaling factors are position-dependent). Besides resulting in position-independent values, this has the advantage that rounding errors are avoided, since for $r = 1$ and $r = 2$ the values of V_{ij} are approximations of the actual values that would be obtained by Eq. (4). As in the original V_{ij} coefficients, we avoid working with fractional values, and multiply the coefficients by a factor of 16, resulting in integer multiplier coefficients (any lower power of 2 would still result in fractional values).

$$V' = \frac{16 \cdot Q_{step}}{2^{\lfloor QP/6 \rfloor}}. \tag{5}$$

In this way, $V' = V_{ij|r=0}$ (i.e., the value which is among others used for the DC coefficient), since $E_{I,ij|r=0} = 1/4$. As mentioned, we want to avoid unnecessary up- and downscaling operations, i.e., so that $M' \cdot V' \gg 15 = 1$ (compare to $M_{ij} \cdot V_{ij} \gg 15 = 64 \cdot E_{F,ij} \cdot E_{I,ij}$). In order to achieve this, the forward multiplier coefficients are updated as follows, by incorporating the factor of 16 used for construction of V':

$$M' = 2^{11+\lfloor QP/6 \rfloor} \cdot \frac{1}{Q_{step}} \, . \tag{6}$$

We obtain the updated position-independent quantizer coefficients as given in Table 1. By using these updated values, redundant scaling operations are eliminated in the requantization process, leading to more accurate requantized coefficients.

Table 1

Modified (position-independent) multiplier coefficients M' and V' for requantization.

$QP\%6$	M'	V'
0	3277	10
1	2979	11
2	2521	13
3	2341	14
4	2048	16
5	1821	18

2.2.2 Successive quantization error

When compared to direct encoding, there may be an additional error since successive quantization gradually decreases the residual information. Successive quantization may result in requantization errors since the second generation quantizer has only access to the first generation quantized transform coefficients instead of the original transform coefficients [17].

Successive quantization consists of a first generation and a second generation quantization. During the first generation quantization, the original transform coefficients W_{ij} are requantized resulting in the values $Z_{ij,1}$. After inverse quantization, the reconstructed coefficients $\widehat{W}_{ij,1}$ are obtained. The second generation quantization takes these values as input and results in the values $Z_{ij,2}$. The final reconstructed values are indicated as $\widehat{W}_{ij,2}$. Ideally, successive quantization should be equivalent to direct encoding: $\widehat{W}_{ij,2} = \widehat{W}_{ij}$. However, in most

cases requantization errors are introduced since direct encoding is different from successive quantization.

In [18], a requantization theorem for uniform scalar quantizers was derived. It showed that 'perfect' requantization can be achieved when the second quantizer is embedded in the original quantizer.

The practical applicability of an open-loop perfect requantization system for H.264/AVC was mentioned in [18]. The problem of drift due to the energy loss and the absence of error compensation was mentioned, but its impact was not studied. In the remainder of this paper, we will show that open-loop requantization leads to significant drift; this also applies to the case of 'perfect' requantization.

2.3 Requantization error drift

Although perfect requantization can be achieved in certain cases, i.e., the output coefficient after transcoding will be identical to the one obtained by direct encoding, the loss of information caused by coarser quantization with QP_2 will accumulate and propagate. In MPEG-2, this was the case due to motion-compensated prediction. In H.264/AVC, however, the source of drift will no longer be restricted to MCP alone. In fact, both spatial and temporal drift will be found because of intra prediction and MCP, respectively.

An efficient version of intra prediction was adopted by the H.264/AVC specification. The efficiency is obtained by exploiting the spatial redundancy between the pixels in the current (4×4 , 8×8 or 16×16) block and the surrounding (reconstructed) pixels of the neighboring (macro)blocks. H.264/AVC intra prediction

results in improved compression efficiency, but also introduces new dependencies in the video bitstream. When requantization is applied to intra-coded macroblocks in H.264/AVC bitstreams (which can be present in all picture types), spatial drift propagation can be noticed. In order to avoid spatial drift, algorithms for requantization transcoding need to be (re)assessed.

Different requantization transcoding solutions have been proposed [19,20], in both pixel and compressed domain. The most straightforward solution for transcoding is the decoder-encoder cascade. In this case, the incoming bitstream is fully decoded, and reencoded using the given parameters (such as a predefined bit rate). Due to its computational complexity, however, this solution is in many cases not feasible. Different reduced-complexity alternatives are available, two of which are the cascaded-pixel domain transcoder (CPDT) and the open-loop transcoder (OL).

In the remainder of this section, we study the drift as found in H.264/AVC transcoding. When compared to MPEG-2, a number of changes lead to a different analysis. In the following discussion, the superscript 1 indicates a decoder-side signal and the superscript 2 denotes an encoder-side signal. Lowercase variables represent pixel-domain signals, while uppercase variables denote the equivalent signal in the transform domain. All notations are presented in Table 2 ($k \in \{1, 2\}$).

2.3.1 Cascaded pixel-domain transcoding

The CPDT architecture can be considered as the reference model for requantization transcoding. This closed-loop architecture is by definition drift-free, but requires more processing power compared to compressed-domain solutions,

Table 2

List of notations.

Symbol	Description
C_n^k	quantized and transformed signal at decoder/encoder side
E_n^k	transformed error signal at decoder/encoder side
e_n^k	error signal at decoder/encoder side
x_n^k	reconstructed signal at decoder/encoder side
y_n^k	reference signal at decoder/encoder side
$Q_i(\cdot)$	quantization (quantization parameter QP_i)
$I_m(\cdot)$	intra prediction (prediction mode m)
$M_{\boldsymbol{v}}(\cdot)$	motion compensation (motion vector $\boldsymbol{v}$)

due to the double prediction loop. Instead of performing motion estimation for the output video bitstream, the mode decisions and motion information of the input video bitstream are reused. In this way, the computational complexity is significantly reduced [2]. The CPDT architecture is depicted in Fig. 3. In the figure, a distinction is made between the buffer for the current picture, which is used to store reconstructed values for intra prediction, and the reference picture buffers (lists L0 and L1), which are used for MCP. In H.264/AVC, only pictures from reference picture list L0 are used for P-type macroblocks, while for B-type macroblocks, a choice can be made between reference pictures

[2] Note that if desired, the MVs and mode information can be refined to better reflect the characteristics of the output video signal, at the cost of increased computational complexity [8].

from list L0, L1, or a (weighted) combination of pictures from lists L0 and L1. After reconstruction of the current picture, the deblocking filter (DF) is applied, after which the picture can be stored for future reference.

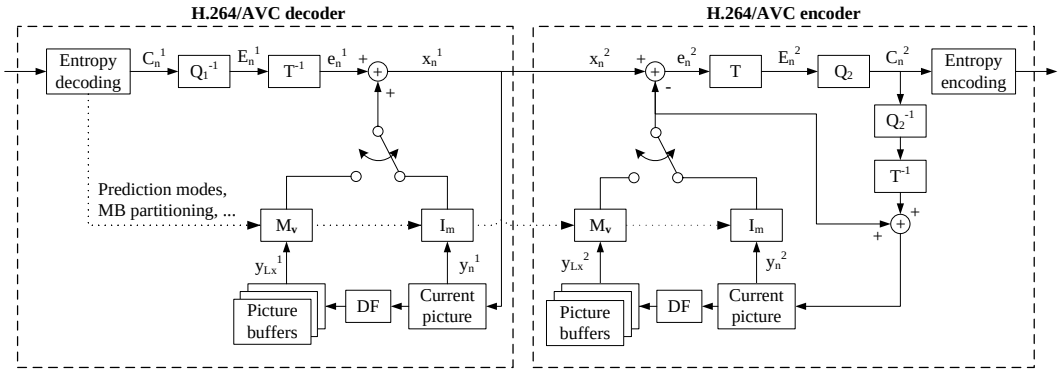


Fig. 3. Cascaded pixel-domain transcoder architecture (CPDT).

For an individual block, the decoded pixels at decoder side and the corresponding prediction error at encoder side are given by the following expressions in case of intra prediction and MCP, respectively:

$$x_n^1 = \begin{cases} I_m(y_n^1) + e_n^1 \\ \\ M_v(y_{\mathrm{Lx}}^1) + e_n^1 \end{cases} \tag{7}$$

and

$$e_n^2 = \begin{cases} x_n^2 - I_m(y_n^2) \\ \\ x_n^2 - M_v(y_{\mathrm{Lx}}^2) \end{cases}, \tag{8}$$

where $I_m(\cdot)$ denotes the intra prediction operator with prediction mode m and $M_v(\cdot)$ represents the motion compensation operator with motion vector v. For simplicity, we omit the second motion vector in case of bidirectionally predicted blocks in B pictures. Since $x_n^1 = x_n^2$, Eq. (7) can be substituted in Eq. (8):

$$e_n^2 = \begin{cases} I_m(y_n^1) + e_n^1 - I_m(y_n^2) \\ \\ M_v(y_{\mathrm{Lx}}^1) + e_n^1 - M_v(y_{\mathrm{Lx}}^2) \end{cases} . \tag{9}$$

The output prediction error equals the addition of the input prediction error and the difference between the prediction signal in decoder and encoder loop. This holds for both intra prediction and MCP.

The input and output prediction errors of frame n are related to input and output quantized coefficients indicated by C_n^1 and C_n^2 respectively:

$$e_n^1 = T^{-1}(Q_1^{-1}(C_n^1)) , \tag{10}$$

and

$$C_n^2 = Q_2(T(e_n^2)) . \tag{11}$$

In this way, the output coefficients C_n^2 before entropy coding can be written as (due to linearity of the core forward integer transform):

$$C_n^2 = \begin{cases} Q_2[T(T^{-1}(Q_1^{-1}(C_n^1))) + T(I_m(y_n^1) - I_m(y_n^2))] \\ \\ Q_2[T(T^{-1}(Q_1^{-1}(C_n^1))) + T(M_v(y_{\mathrm{Lx}}^1) - M_v(y_{\mathrm{Lx}}^2))] \end{cases} . \tag{12}$$

In case all coding parameters, i.e., mode decisions and motion information, remain the same for the output video bitstream, the above expressions can be simplified by merging the input and output coding loops. This, however, assumes linearity of the intra prediction and MCP processes, which is not the case in general (due to non-linear operations, as will be discussed in Sect. 2.3.5):

$$C_n^2 \approx \begin{cases} Q_2[T(T^{-1}(Q_1^{-1}(C_n^1))) + T(I_m(y_n^1 - y_n^2))] \\ Q_2[T(T^{-1}(Q_1^{-1}(C_n^1))) + T(M_v(y_{\mathrm{Lx}}^1 - y_{\mathrm{Lx}}^2))] \end{cases} . \quad (13)$$

2.3.2 Open-loop transcoding

The OL transcoder is the simplest solution for requantization, but it is also characterized by severe drift propagation. The residual information is dequantized with QP_1 and requantized with QP_2 (typically, $QP_1 < QP_2$). The OL transcoder is shown in Fig. 4. Note that the modified multiplication coefficients M' are reflected in the notation Q_2' for the requantization step.

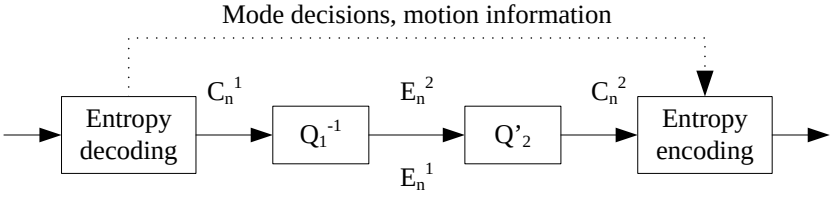


Fig. 4. Open-loop transcoder.

The inverse quantization can be described as

$$E_n^1 = Q_1^{-1}(C_n^1) \ , \quad (14)$$

while the requantization can be formulated as

$$C_n^2 = Q_2'(E_n^2) \ , \quad (15)$$

Since $E_n^1 = E_n^2$ for the OL transcoder, Eq. (14) and Eq. (15) can be combined:

$$C_n^2 = Q_2'(Q_1^{-1}(C_n^1)) \ . \quad (16)$$

It should be mentioned that, apart from losses in the inverse transform, the following expression applies, where the adapted multiplication factors for the

requantization process are used for requantization Q'_2:

$$Q_2(T(T^{-1}(Q_1^{-1}(\cdot)))) \approx Q'_2(Q_1^{-1}(\cdot))) \ . \tag{17}$$

In this way, we can write the above expression (Eq. (16)) as follows, reintroducing the inverse and forward transform processes in order to obtain the relation between the input and output coefficients in the OL transcoding solution:

$$C_n^2 \approx Q_2\Big[T(T^{-1}(Q_1^{-1}(C_n^1)))\Big] \ . \tag{18}$$

2.3.3 Drift components

By comparing Eq. (13) to Eq. (18), two drift components can be identified. These two terms constitute the difference in E_n^2 between the CPDT and OL solutions. In the case of intra prediction, a spatial drift term D_s is obtained:

$$D_s = T(I_m(y_n^1 - y_n^2)) \ . \tag{19}$$

For motion-compensated prediction, a temporal drift term D_t is identified:

$$D_t = T(M_v(y_{\mathrm{Lx}}^1 - y_{\mathrm{Lx}}^2)) \ . \tag{20}$$

The two components reinforce each other, since intra-predicted blocks can be used as reference for MCP, and vice versa.

Temporal drift component: The temporal drift component results from requantization errors that propagate in the motion compensation loop. Because of motion-compensated prediction, reference pictures will be affected, leading to an incorrect prediction signal. As a result, quality will decline due to both the affected reference used for prediction and the coarser quantization in the current picture. This type of drift also had to be dealt with in transcoded

MPEG-2 streams. In H.264/AVC, multiple (long-term) reference picture MCP [21] is applied, which allows errors to propagate beyond the boundaries of the current GOP. In this way, the temporal drift accumulates and propagates from (reference) frame to frame until an IDR picture is processed.

Spatial drift component: Since spatial dependency coding is not present in MPEG-2, this type of drift did not occur in MPEG-2 transcoding. Spatial drift accumulates and propagates from block to block according to the intra prediction modes. Since intra-predicted macroblocks can occur in I, P, and B pictures, this type of drift has a significant impact in all picture types. In particular, in high-motion regions in P and B pictures, intra macroblocks are often inserted as the rate-distortion-optimal choice by the encoder. Also, when the distance between reference pictures increases, intra prediction often becomes the optimal choice during encoding. This is for example the case when using hierarchical GOP structures, in the lowest temporal layers (as for example in SVC [22]).

2.3.4 Drift visualization

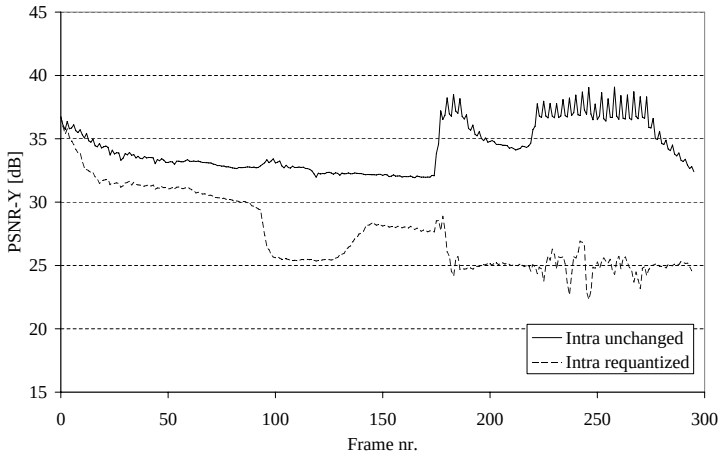
The importance of the spatial drift term can be seen in Fig. 5(a), where the PSNR values of the Stefan sequence (CIF resolution) are shown frame per frame after transcoding. A long IBBP GOP structure was used (298 frames, only the first frame is intra-coded) to also visualize the temporal drift effect. Drift is avoided in the first (intra-coded) picture by using the double-loop CPDT architecture for this frame.

The top curve illustrates the case where intra macroblocks in P and B pictures are not requantized, while the P and B macroblocks are requantized open-loop.

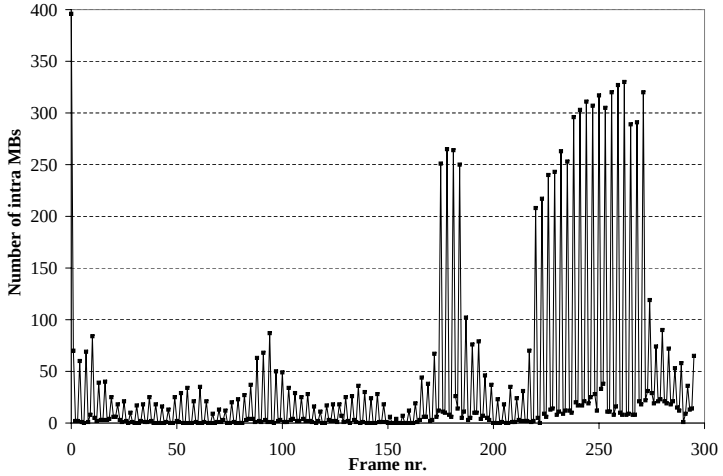
An increase of the quantization parameter by 4 is used (from QP 22 to QP 26). Due to the high motion content of the Stefan sequence, temporal drift can be noticed. By additionally requantizing the intra-coded macroblocks for the same sequence, the bottom curve is obtained (in this case, all macroblocks in P and B pictures in the stream are open-loop transcoded). For this curve, the impact of spatial drift becomes visible. The variations in the PSNR values are related to the amount of intra-coded macroblocks as shown in Fig. 5(b). This graph shows the amount of MBs that were coded as intra macroblock in the input video stream (out of a total of 396 macroblocks per frame). It can be seen that when the amount of intra MBs increases in a frame, the quality for the lower curve will decrease significantly when compared to the top curve. Because of spatial drift, quality can decrease within a single frame, while for temporal drift, quality slowly deteriorates over a number of frames. This can be seen intuitively, given the high number of dependencies in intra-coded macroblocks [16].

In contrast, for the top curve, more texture information is preserved by leaving the intra-coded macroblocks unchanged, leading to higher PSNR values when large intra-coded regions are present in the frames (which keep their initial quality at $QP = 22$. This effect becomes visible in the second half of the sequence.

In the proposed architectures in the following sections, temporal and spatial compensation loops are introduced to tackle both drift terms. Requantization with spatial compensation will also be applied to intra-coded macroblocks to obtain a uniform quality in the stream (hereby avoiding the peaks as shown in the second half of Fig. 5(a).



(a) Sequence without and with spatial drift (Stefan, CIF resolution).



(b) Number of intra MBs per frame (Stefan, CIF resolution).

Fig. 5. Comparison of the quality of frames in the Stefan sequence without and with requantization of intra-coded macroblocks ($QP_1 = 22$, $QP_2 = 26$).

Since spatial drift propagation results in annoying block artifacts in the decoded frames, precautions should be taken in order to avoid this kind of drift. Typical artifacts are visualized in Fig. 6 and Fig. 7. Fig. 6(a) shows a frame from the Stefan sequence, transcoded using the CPDT architecture. Fig. 6(b)) shows the same frame, transcoded open-loop. Drift is clearly visible, and propagates throughout the intra-coded frame, according to the intra prediction

modes.

Spatial drift is not limited to intra-coded pictures, however, since intra-coded macroblocks can also be inserted in P and B pictures. This is visualized in Fig. 7(a) and Fig. 7(b)). In the latter, requantization errors propagate in intra-coded regions. When these macroblocks are used as reference for MCP, these drift errors propagate to depending frames. This results in severe drift errors, with worsening artifacts towards the end of GOPs.

2.3.5 Non-linear operations

A number of non-linear operations prevent the formulas derived above from being exact, resulting in the approximation of the drift components in Eq. (19) and Eq. (20).

- **Intra prediction** The H.264/AVC specification provides nine intra 4×4 and four intra 16×16 prediction modes for the luma component, and four modes for the chroma components. Besides the horizontal and vertical prediction modes, a DC prediction mode and diagonal prediction modes are provided. The latter modes make use of divisions in order to calculate their prediction values. These integer divisions are non-linear operations, which may result in arithmetic rounding errors.
- **Sub-pixel interpolation** When motion vectors in H.264/AVC refer to sub-pixel displacement positions, the sub-pixel position values are obtained by interpolation. For half-pixel displacements, a 6-tap filter is applied with filter coefficients $(1, -5, 20, 20, -5, 1)/32$. For quarter-pixel accuracy displacements, the half-pixel values are additionally interpolated using a bilinear filter. When applying the interpolation formulas to the requantization dif-

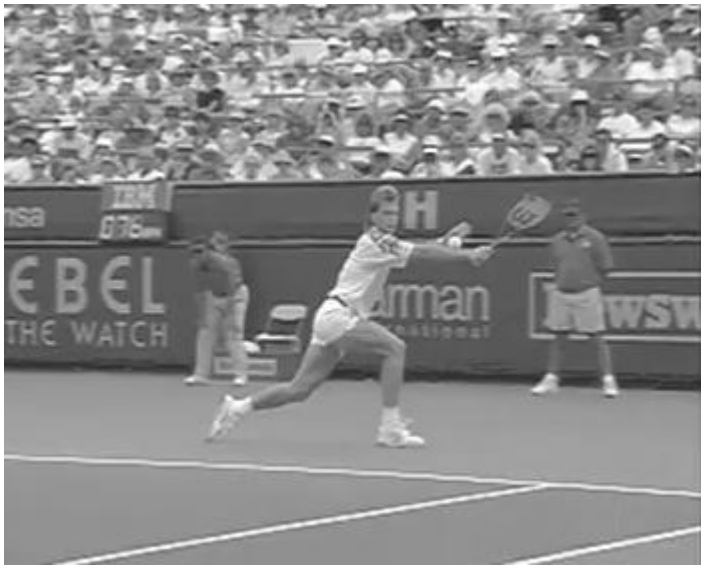
(a) Driftless transcoded intra-coded picture (Stefan, frame 15, transcoded using CPDT, $QP_1 = 22$, $QP_2 = 28$)

(b) Spatial drift in intra-coded picture (Stefan, frame 15, transcoded using OL, $QP_1 = 22$, $QP_2 = 28$)

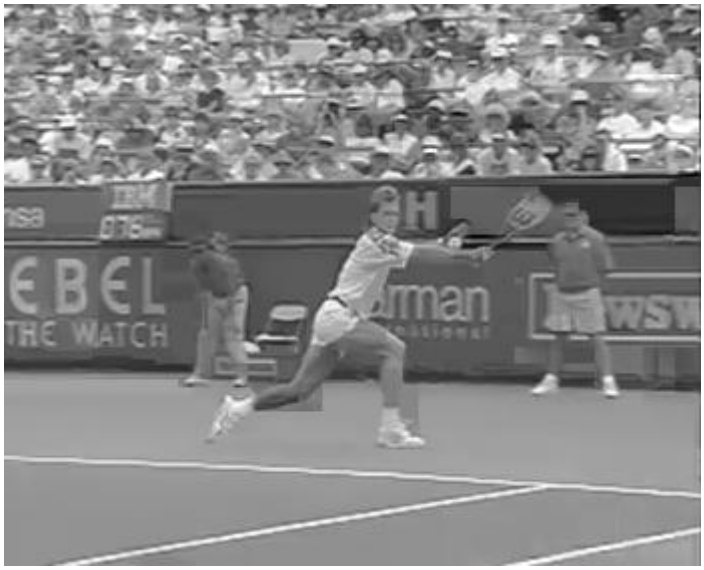
Fig. 6. Spatial drift propagation in intra-coded pictures caused by OL transcoding (Fig. 6(b)) vs. driftless transcoding with CPDT (Fig. 6(a)).

ference values in the reference frames, rounding errors may arise caused by integer arithmetic.

- **Inverse transform** The inverse integer transform in the H.264/AVC speci-



(a) Driftless transcoded P picture (Stefan, frame 183, transcoded using CPDT, $QP_1 = 22$, $QP_2 = 28$)



(b) Spatial drift in intra MBs in P picture (Stefan, frame 183, transcoded using OL, $QP_1 = 22$, $QP_2 = 28$)

Fig. 7. Spatial drift propagation in intra-coded MBs in P pictures caused by OL transcoding (Fig. 7(b)) vs. driftless transcocding with CPDT (Fig. 7(a)).

fication operates on values which are multiplied by a factor of 64 in order to prevent precision losses during the transform (see Eq. (4)). After the inverse transform, the values have to be downscaled again in order to correspond

to the magnitude of the original values. This downscaling process results in rounding errors as a consequence of the division (shift operation).

- **Bit depth clipping operations** A cascaded decoder-encoder solution will completely decode the video sequence, resulting in pixel values in the range of 0 to 255 (for a bit depth of 8 bits). In certain cases, these boundaries may be crossed during compensation (intra prediction or motion compensation), and clipping is necessary to the boundaries of the range. Since reconstructed values are not available in the open-loop transcoder, clipping cannot be performed, and may result in drift.
- **Deblocking filter** In the CPDT architecture, H.264/AVC in-loop deblocking filtering [23] is applied, resulting in altered values in the reconstructed pictures. This operation is not performed in the open-loop transcoder. In order to overcome block artifacts in the transcoded sequence, techniques for deblocking in the transform domain could be applied, as in [24] and [25].

In the following section, we investigate algorithms that reduce the temporal and spatial drift components, while maintaining low complexity of the transcoder. Otherwise stated, we examine architectures that improve rate-distortion performance of the open-loop transcoder, at complexity lower than the CPDT transcoder. Architectures that refine mode information and motion vectors, hence having higher complexity than the CPDT transcoder, have been studied in among others [8] and [9].

3 Single-loop architectures with spatial and/or temporal compensation

3.1 Basic single-loop requantization transcoder with temporal compensation

One approach for simplification of the CPDT transcoder is to identify and combine the decoder and encoder loop and to merge common modules based on the assumed linearity of the transform and motion compensation [26]. In this way, a simplified 'single-loop' architecture is obtained. The traditional single-loop transcoder architecture, which was used for MPEG-1/2, is shown in Fig. 8. When compared to the open-loop transcoder, a compensation loop is added where the difference between the original coefficients and the requantized coefficients is used for compensation, corresponding to the drift term D_t in Eq. (20). The compensation prevents that errors, which result from the requantization process, propagate to depending blocks. This architecture was used for MPEG-1/2, for temporal compensation of requantization errors. The depicted architecture performs motion compensation in the pixel domain, hence the need for an inverse transform. A transform-domain approach was also used for MPEG-2, with a DCT-domain motion compensation module. This was discussed for example in [6].

3.2 Single-loop architectures with spatial compensation

In literature, fast architectures for H.264/AVC transcoding are mostly based on the CPDT architecture, while single-loop architectures were mentioned as not to be useful in practical applications [8,9,15]. These results were based

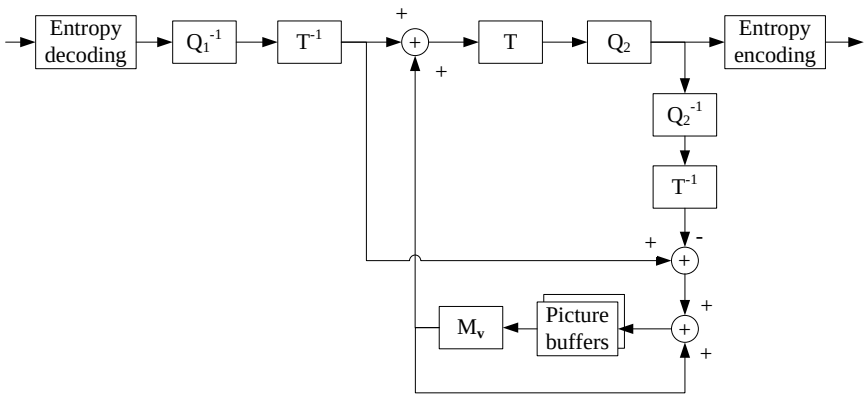


Fig. 8. General architecture of the 'traditional' single-loop (pixel-domain) requantization transcoder with temporal compensation.

on the observation of uncontrollable drift in intra-coded macroblocks, and did not take into account the drift term D_s in Eq. (19). Here we present a single-loop architecture which does take into account spatial drift propagation. In this way, single-loop architectures become practically viable, leading to a new class of transcoders available for H.264/AVC transcoding.

Since no form of intra prediction (except for DC coefficients in intra-coded pictures) was present in MPEG-2 Video, the existing architecture can no longer be used as such for transcoding of H.264/AVC sequences. As demonstrated in the previous sections, compensation of spatially propagating errors is indispensable. This leads us to introduce an H.264/AVC requantization transcoder with spatial compensation, as shown in Fig. 9. In this architecture, a spatial compensation loop is used. A buffer is maintained for the current picture, which is used for compensation of intra-predicted macroblocks. The buffer stores (accumulated) requantization errors. These values are used to form a spatial compensation signal for the neighboring macroblocks, according to the used intra prediction modes.

To tackle both spatial and temporal drift terms, this architecture can be

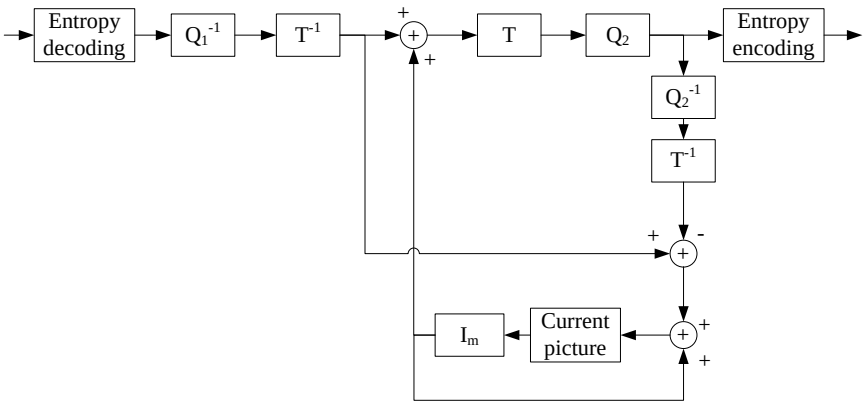


Fig. 9. H.264/AVC requantization transcoder with spatial compensation.

extended to the hybrid requantization transcoding architecture as shown in Fig. 10. This scheme was proposed by the authors in [27], and can be applied to P and B pictures containing both MCP and intra-predicted macroblocks. In this paper, we further extend the use of this technique to combined architectures which provide a trade-off between computational complexity and rate-distortion performance. Depending on the macroblock type, a different type of compensation is used.

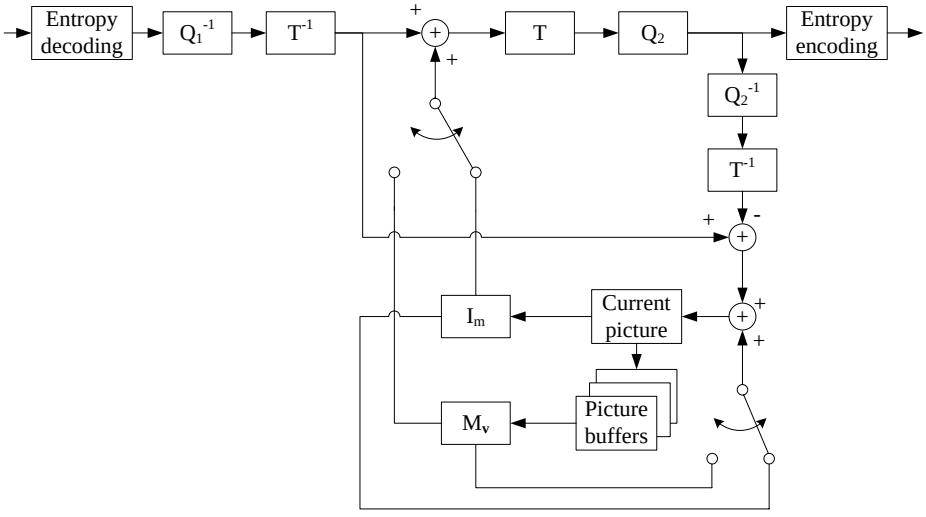


Fig. 10. H.264/AVC hybrid requantization transcoder (spatial and temporal compensation).

Since multiple reference frames can be used in H.264/AVC, a reference frame

buffer is constructed as at the decoder side, containing pictures with requantization errors. After transcoding each frame, the content of the current frame buffer is added to the reference frame buffer. Since we are working on residual error coefficients, deblocking is not performed in the transcoder.

3.3 Transform-domain optimizations

A number of optimizations can be made to the presented architectures, by shifting operations back to the transform domain. Transform-domain solutions that were applicable to MPEG-2 cannot be applied as such in H.264/AVC.

The possibility of using a pure transform-domain solution for intra prediction was discussed in [28]. This, however, resulted in formulas requiring a significant amount of floating-point operations, with high complexity for most of the intra prediction modes (in particular 4×4 modes 3 to 8). Because of rounding errors, transform-domain intra prediction turns out not to be useful in practical transcoding situations.

For the same reasons (rounding errors and complexity), motion compensation is still performed in the pixel domain, whereas for MPEG-2, MC-DCT was used to speed up transcoding. In [29], an algorithm was discussed for motion compensation in the transform domain, used in the context of MPEG-2 to H.264/AVC transcoding. The authors, however, did not take into account the sub-pixel interpolation process. In this way, the algorithm could only be used for full-pixel accuracy. Adjusting the algorithm to quarter-pixel accuracy, and the introduced dependencies would greatly increase the computational complexity. Together with the introduced rounding errors, this would render

the transform-domain approach useless.

Another solution is to use a partial transform-domain architecture, as shown in Fig. 11 for the hybrid architecture. One inverse transform is still used, and motion compensation is applied in the pixel domain. A number of simplifications can be implemented.

3.3.1 Intra prediction

For the intra prediction, efficient compensation formulas can be obtained with lower complexity than the combination of intra prediction and forward transform. In particular, sparse compensation matrices are obtained for the 16×16 prediction modes and for intra 4×4 modes 0 to 4. For these modes, the combination of intra prediction and transform leads to lower complexity. The combination is indicated by the dashed line in Fig. 11. The other 4×4 prediction modes (5 to 8) result in compensation matrices that are less sparse and require a larger amount of multiplications. For these modes, it is more beneficial to apply traditional pixel-domain intra prediction, followed by the forward integer transform (instead of a combined approach). Compensation matrices for the 4×4 and 16×16 prediction modes and the impact on the performance of requantization transcoders by using these techniques are discussed in [30,31].

3.3.2 Motion-compensated prediction

A similar technique could be applied by forming a combination of motion compensation and forward transform. This, however, results in an increased number of calculations, and in particular multiplications, required for this operation, when compared to the 16×2 multiplications (2 multiplications per

fractional pixel) required in the pixel-domain interpolation formulas, given the 6-tap filter with symmetric weights $(1, -5, 20, 20, -5, 1)/32$.

Since interpolation is performed on inverse transformed (accumulated) requantization error values, the question is if it is beneficial to use a relatively complex 6-tap Wiener filter during subpixel interpolation. In [32,33], the problem of aliasing is mentioned as the main reason for using a Wiener filter with 6 or 8 taps instead of a bilinear filter. The 6-tap H.264/AVC interpolation filter is based on the separable 2-D Wiener filter proposed in [33], which was used to reduce drift caused by aliasing for multiresolution hybrid video coding.

Complexity can be reduced, however, by using a bilinear filter instead of a 6-tap filter. When compared to the 6-tap Wiener filter, a bilinear filter has higher passband attenuation and stopband permeability, resulting in a less accurate low-pass filter. Apart from arithmetic differences between the 6-tap and bilinear filters, the use of a bilinear filter will lead to more aliasing in the predicted signal for sub-pixel positions. If low complexity transcoding is required, however, the use of a bilinear interpolation filter could be considered in order to reduce complexity, as we discussed in [34].

3.4 Overall architectures

Since the spatial and temporal compensation is not perfect due to non-linear operations, some errors can still accumulate and propagate. This is especially troublesome for intra-coded macroblocks, which contain a high number of dependencies (even within a single macroblock due to 4×4 intra prediction). In intra-coded pictures (consisting entirely out of intra-coded macroblocks), the

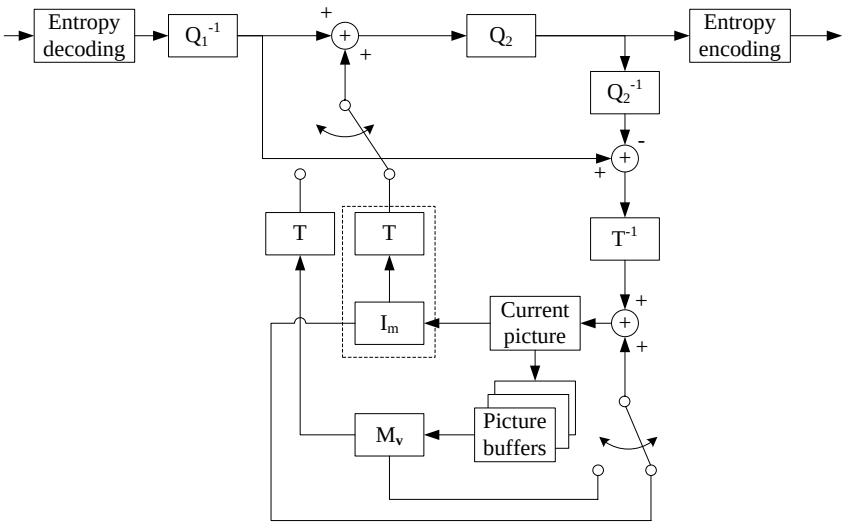


Fig. 11. Partial transform-domain hybrid requantization transcoder.

situation is the most severe. For optimum transcoding performance, 'mixed' architectures can be derived. By using a cascaded decoder-encoder architecture for intra-coded pictures, more reliable (drift-free) reference frames are formed. In [15], this possibility was examined. There, a mixed requantization architecture (MRA) was discussed but resulted in unreliable results, due to the absence of spatial compensation for intra-coded macroblocks in P and B pictures.

By using a combination of spatial and/or temporal compensation, the overall quality is significantly improved. In the transcoder, depending on the picture and/or macroblock type, a different processing technique can be selected. Different overall architectures are hence obtained. The use of combined architectures turns out to be a powerful technique for trading off complexity and rate-distortion performance.

The overall architectures are determined by combining techniques for the individual picture/macroblock types.

- **I pictures.** Intra-coded pictures have the benefit that reconstruction to the pixel domain can be executed with relatively low computational requirements. Single-loop compensation or open-loop transrating can also be applied, but result in reduced quality. Hence, given the relatively low extra computational cost, using the CPDT will be preferred for most applications.
- **Intra-coded macroblocks in P and B pictures.** Reconstruction of intra-coded macroblocks in P and B pictures has the advantage of being drift-free, but requires reconstruction of the surrounding macroblocks, in order to have the prediction pixels available for intra prediction. If one or more of the surrounding macroblocks uses MCP, a dependency on temporally predicted macroblocks is introduced, and reconstruction of MCP macroblocks becomes necessary [3]. Spatial compensation of these intra-coded macroblocks, however, can be applied without imposing reconstruction of surrounding MCP macroblocks, and results in highly improved quality over open-loop requantization.
- **MCP macroblocks.** For MCP macroblocks, a choice can be made between CPDT reconstruction (if the blocks upon which the current block depends are also reconstructed), temporal compensation, or open-loop transrating. Adding temporal compensation or CPDT reconstruction introduces extra computational complexity.

Note that an even finer categorization can be applied, e.g., based on the used GOP structure. For the example case of hierarchical coding structures, pictures that are located lower in the temporal hierarchy are eligible for higher-complexity temporal compensation or pixel-domain reconstruction, while pic-

[3] Note that this is not the case when the *constrained intra prediction flag* is enabled in the active picture parameter set of the H.264/AVC stream.

tures in the highest temporal layers can be transrated open-loop with little effect on the quality of the output video.

By taking into account the restrictions and remarks mentioned above, the following overall architectures are obtained. As a reference, CPDT can be applied to all picture and macroblock types. In the remainder of this chapter, this architecture will simply be referred to as 'CPDT' for brevity. The architecture which uses open-loop transrating for all blocks is denoted as 'OL'.

By using the CPDT architecture for intra-coded pictures, spatial compensation for the remaining intra macroblocks (in P and B pictures), and open-loop requantization for MCP macroblocks, the MRA-SC architecture is obtained (mixed requantization with spatial compensation). Based on this distinction of the input picture and/or macroblock type, a switch is made between the the CPDT architecture shown in Fig. 3, the single-loop architecture with spatial compensation (Fig. 9), and the open-loop transcoder (Fig. 2). MRA-SC benefits from the drift-free reference intra-coded frames created by the CPDT architecture. A full pixel-domain reconstruction is performed for these pictures, with a closed prediction loop at decoder and encoder side. After the intra-coded picture has been transcoded, the reconstructed frames (at decoder and encoder side) are no longer required for future reference, and the memory buffers of the CPDT architecture can be cleared. By using spatial compensation in P and B pictures, MRA-SC tackles the largest source of drift in these pictures, i.e., spatially propagating errors. For intra-coded (macro)blocks, requantization error values from neighboring intra-coded blocks are used to form the prediction signal. Only the bottommost and rightmost pixels of every block can be used for future compensation, which limits the amount of storage needed per 4×4 block to seven pixel values. MCP blocks in P and B

pictures are transcoded open-loop, and only require an inverse and forward quantization step. The MRA-SC architecture will prove to provide significantly improved rate-distortion performance over open-loop transrating at a relatively low computational complexity cost.

The MRA architecture which incorporates both spatial and temporal compensation (hybrid compensation) will be referred to as MRA-Hybrid. Here also, the CPDT architecture is used for intra-coded pictures. The difference between the reconstruction of I pictures at decoder side and encoder side is added to the reference picture buffer for future temporal compensation. This is illustrated in Fig. 12. Furthermore, after transcoding of reference P and B pictures, the corresponding requantization error pictures are added to the temporal picture buffers (i.e., passed from the current picture buffer to the picture buffers for temporal compensation). Spatial compensation for intra-coded macroblocks in P and B pictures is applied in the same way as for MRA-SC, based on the error values from neighboring (macro)blocks, as stored in the current picture buffer. Furthermore, the requantization errors from reference pictures (stored in the picture buffers) are used for temporal compensation.

The more traditional architecture with only temporal compensation is indicated as MRA-TC (MRA with temporal compensation) in the results section. The techniques used for the 'CPDT', 'OL', 'MRA-TC', 'MRA-SC', and 'MRA-Hybrid' architectures are summarized below.

CPDT architecture

- CPDT for intra-coded pictures.
- CPDT for MCP macroblocks.

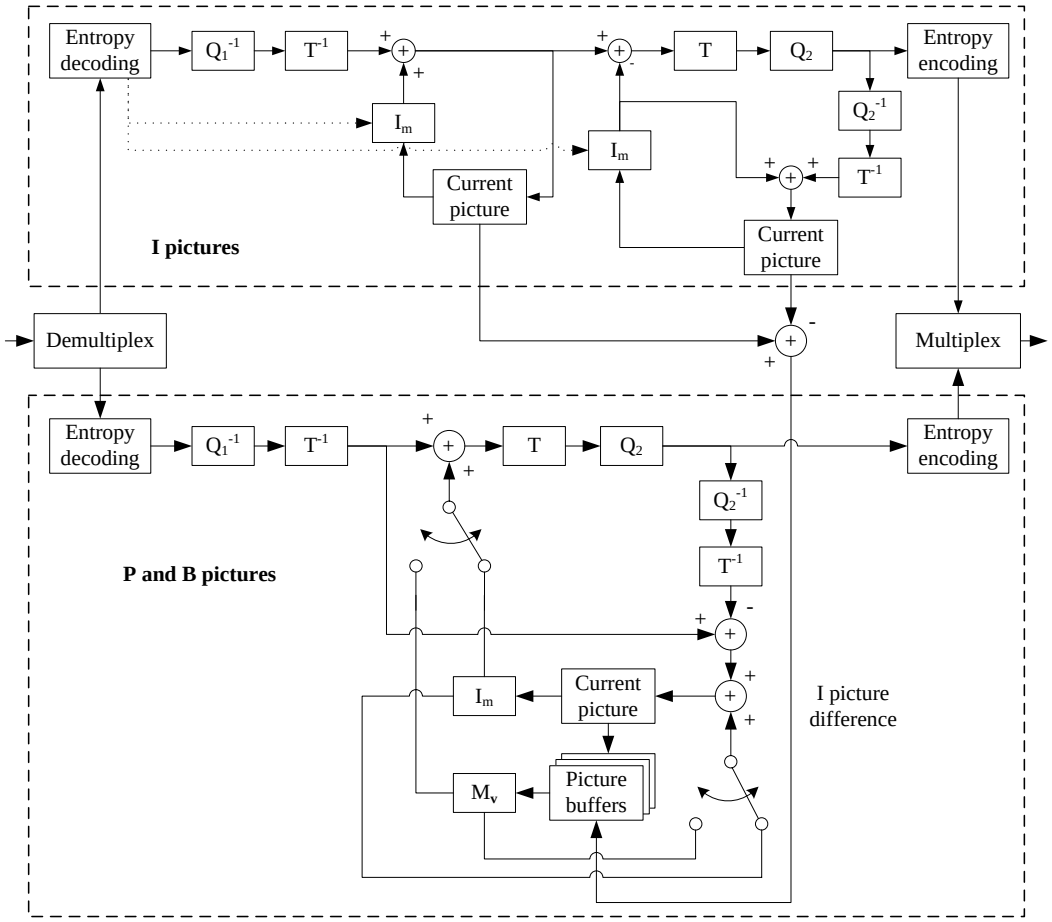


Fig. 12. MRA-Hybrid transcoder architecture.

- CPDT for intra-coded macroblocks in P and B pictures.

OL architecture

- Open-loop for intra-coded pictures.
- Open-loop for MCP macroblocks.
- Open-loop for intra-coded macroblocks in P and B pictures.

MRA-SC architecture

- CPDT for intra-coded pictures.

- Open-loop for MCP macroblocks.
- Spatial compensation for intra-coded macroblocks in P and B pictures.

MRA-TC architecture

- CPDT for intra-coded pictures.
- Temporal compensation for MCP macroblocks.
- Open-loop for intra-coded macroblocks in P and B pictures.

MRA-Hybrid architecture

- CPDT for intra-coded pictures.
- Temporal compensation for MCP macroblocks.
- Spatial compensation for intra-coded macroblocks in P and B pictures.

Experiments describing the impact of the different techniques of these overall architectures are given in the results section.

3.5 Complexity discussion

The complexity of the overall architecture is determined by the techniques applied to the individual picture/macroblock types, i.e., intra-coded pictures, intra-coded macroblocks in P and B pictures, and MCP macroblocks.

Using CPDT for intra-predicted pictures results in a minor increase in complexity of the overall architecture. As will be seen in the results section (Sect. 4.2), adding spatial compensation for intra-coded macroblocks in P and B pictures has only limited impact on the computational complexity.

Adding temporal compensation, however, requires a number of time-consuming operations in H.264/AVC.

- **Motion vector derivation** In H.264/AVC, median motion vector prediction is included, based on the motion vectors of surrounding (sub)macroblock partitions. This context-adaptivity strongly increases complexity when decoding motion vectors.
- **Interpolation** For quarter-pixel accuracy MCP, half-pixel values are obtained by using the 6-tap interpolation filter. This can be a complex operation, in particular for bipredictionally MCP blocks, with MVs pointing to half- or quarter-pixel positions.
- **Reference picture management** A number of algorithms have to be executed for reference picture management, such as decoded reference picture marking and reference picture list reordering. The use of multiple reference pictures for MCP severely increases memory requirements.

4 Performance analysis

4.1 Rate-distortion results

In order to evaluate the rate-distortion performance of the discussed architectures, we transcoded sequences with varying characteristics and spatial resolution: Foreman (QCIF), Stefan (CIF resolution), Crew (4CIF resolution), and Stockholm (720p). The sequences were encoded using the H.264/AVC Joint Model reference software (version 13.2) using default coding tools, Main profile, 5 reference pictures, CABAC entropy coding, and full rate-distortion optimization enabled. The deblocking filter is enabled during encoding for all

sequences. Two GOP structures were used, one with IBBP GOP (length 15 pictures), the other with hierarchically coded pictures (length 16 pictures). The sequences were encoded with starting QP_I values (for I slices) 22, 27, 32, and 37, $QP_P = QP_I + 1$ (P slices) and $QP_B = QP_I + 2$ (B slices). These correspond to the values used in the VCEG common test conditions [35]. In the remainder of the section, the notation QP_1 corresponds to the starting QP for the I pictures.

The proposed transcoder architectures were implemented in software, and were used to generate transcoded streams. The output streams were generated using increasing quantization parameters, i.e., $QP_2 = QP_1 + i$, with $i = 1, \ldots, 6$, for all slice types. By using fixed increases of the QP, we eliminate the impact of rate control algorithms in the performance of the transcoder algorithms.

For all shown results, the luma PSNR (PSNR-Y) values are given. Nonetheless, the techniques used for the luminance channel are applied to the chroma channels as well. Since less drift is noticed for the chroma channels (due to 4:2:0 subsampling), the luma channel poses the biggest challenge and its results are presented as a 'worst-case scenario'.

In Fig. 13, rate-distortion results are shown for the Stefan sequence. It is clear that open-loop transcoding is not usable for H.264/AVC. The loss of information due to requantization results in unpredictable drift, which is driven by the combination of temporal and, in particular, spatial prediction. By using the MRA-TC architecture (CPDT architecture for intra-coded pictures, and temporal compensation for MCP blocks), output quality is already significantly improved relative to open-loop transcoding. Still, a gap of about 3 to 4 dB exists when compared to CPDT transcoding, corresponding to the results in

[15].

Remark that a large gap in bit rate occurs between the R-D points for $\Delta QP = 3$ and $\Delta QP = 4$, in particular for the open-loop transcoder. This is explained by the removal of small coefficients from the bitstream, when a given threshold of ΔQP is crossed during requantization. In particular, for typical values of $\epsilon = 1/3$ (intra coding) or $1/6$ (inter coding), requantization of coefficients with absolute value equal to one results in

$$Q_2'(Q_1^{-1}(1)) = 1 \tag{21}$$

for $\Delta QP \leq 3$, while

$$Q_2'(Q_1^{-1}(1)) = 0 \tag{22}$$

for $\Delta QP \geq 4$. It can be shown that this property holds irrespective of the starting QP_1, for the given ϵ value. Due to the accurate prediction in H.264/AVC, residual data consists to a large extent out of coefficients with small absolute value (in particular 1 and -1). When these coefficients disappear from the bitstream, a large decrease in bit rate will occur.

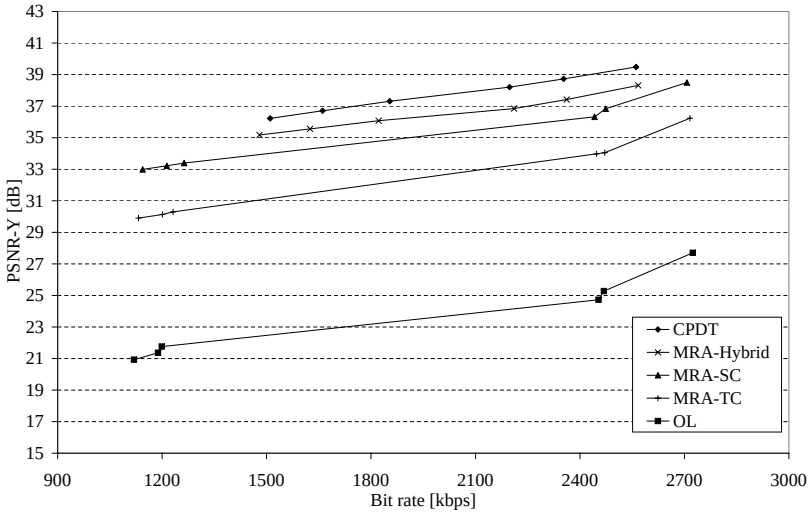


Fig. 13. Rate-distortion results (Stefan, CIF, 30Hz, $QP_1 = 22$).

When we look at visual results, distinct visual artifacts still appear in MRA-TC-transcoded pictures containing intra-coded macroblocks or regions (see Fig. 14(a) and Fig. 15(a)). Applying spatial compensation (MRA-SC) resolves this issue, and improves objective quality by 1 to 2 dB. Since only intra prediction is required for spatial compensation, and MCP is avoided (MCP is significantly more complex than intra prediction, among others due to the interpolation process), MRA-SC can be regarded as a low-complexity transcoder solution. The hybrid architecture with spatial and temporal compensation (MRA-Hybrid) results in additional gain of about 0.5 to 1 dB, depending on the motion characteristics of the sequence. When looking at visual results, artifacts are removed in the pictures, as can be seen from Fig. 14(b) and Fig. 15(b).

For low-motion sequences, the benefit (in a rate-distortion sense) of adding temporal compensation to the MRA-SC architecture vanishes for higher bit rate reductions ($\Delta QP \geq 4$), as can be seen from the detail (the open-loop graph has been omitted) in Fig. 16. Here it can be seen that, although temporal compensation in MRA-Hybrid improves the PSNR values for corresponding R-D points (with identical ΔQP), the amount of residual data increases rapidly for MRA-Hybrid (due to the re-introduction of small residual coefficients). The rate-distortion graph locally drops below that of the MRA-SC architecture, resulting in superior rate-distortion performance of the MRA-SC architecture.

Crew is a particularly challenging sequence, given the high amount of luminance changes in the images (due to the flash lights in the video sequence). Because of this reason, a high number of intra macroblocks is inserted. Still, the MRA-Hybrid architecture is able to restrict the quality loss to 1-2 dB when compared to the decoder-encoder cascade.

(a) Spatial drift in Foreman sequence when using MRA-TC architecture (Foreman, CIF, frame 138, $QP_1 = 22$, $QP_2 = 26$)

(b) Spatial drift is removed after adding spatial compensation (MRA-Hybrid) (Foreman, CIF, frame 138, $QP_1 = 22$, $QP_2 = 26$)

Fig. 14. Transcoded P pictures without and with spatial compensation (Foreman).

(a) Spatial drift in Stefan sequence when using MRA-TC architecture (Stefan, CIF, frame 177, $QP_1 = 22$, $QP_2 = 26$)

(b) Spatial drift is removed after adding spatial compensation (MRA-Hybrid) (Stefan, CIF, frame 177, $QP_1 = 22$, $QP_2 = 26$)

Fig. 15. Transcoded P pictures without and with spatial compensation (Stefan).

Results for the Stockholm sequence (720p) are shown in Fig. 18, and are similar to the results for the lower resolutions. A gap of about 1 dB is found for the

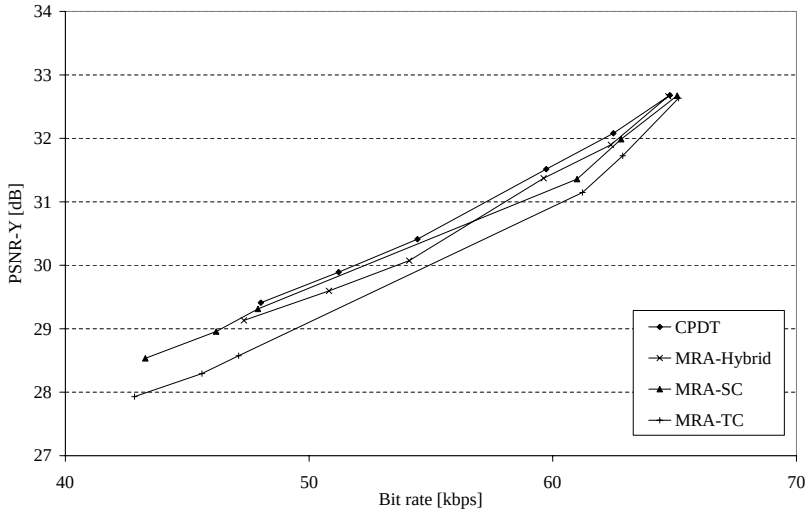


Fig. 16. Rate-distortion results (Foreman, QCIF, 30Hz, $QP_1 = 32$).

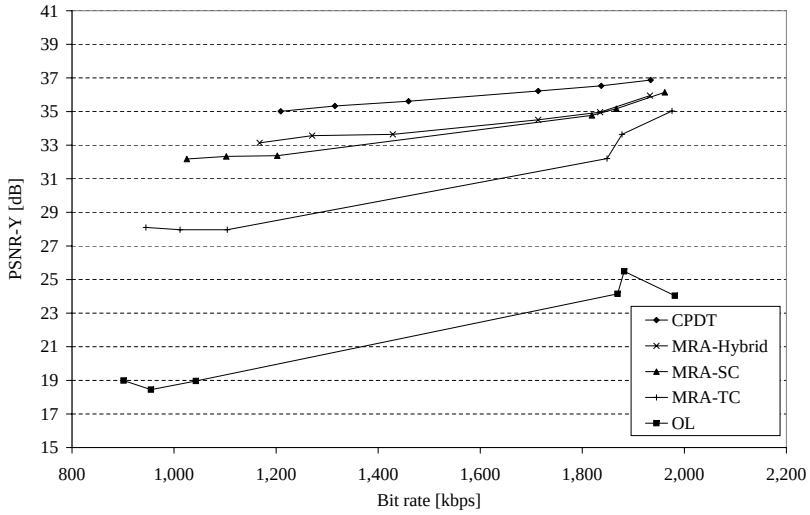


Fig. 17. Rate-distortion results for transcoding architectures (Crew, 4CIF, $QP_1 = 27$).

MRA-Hybrid architecture when compared to the cascaded decoder-encoder approach.

Remark that quality can be further improved by dynamically changing the architecture. If high(er)-quality transcoded sequences are required, the transcoder might choose not to requantize the intra-coded macroblocks. In this way, drift will only propagate temporally, and pictures with a high amount of intra-

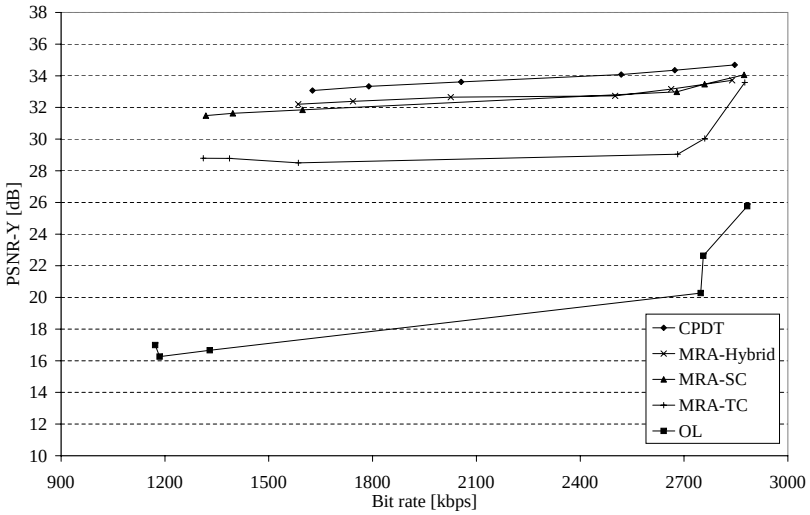


Fig. 18. Rate-distortion results (Stockholm, 720p, $QP_1 = 27$).

coded macroblocks will retain the highest achievable quality. This, however, impacts the rate control ability of the transcoder. If a certain target bit rate is to be achieved, this would imply that the remaining MCP blocks would have to be requantized even more coarsely. Since all intra-coded macroblocks were requantized in the tests described above, the results can be considered a worst case scenario, and a lower bound for such adaptive requantization architectures. Further work might study such adaptive architectures with advanced rate control algorithms for optimum performance.

By looking at the PSNR values over a series of frames, the impact of the individual techniques turns visible. Fig. 19 shows the PSNR values over a sequence of 30 frames for the Stockholm sequence. Since the CPDT architecture is used for I pictures for all four displayed architectures, the PSNR values coincide at the GOP boundaries (GOP of 15). From the first predictive (P/B) pictures, however, the impact of spatial drift can be noticed. For this sequence containing little motion, a quality loss of between 0.2 dB and 0.4 dB is already noticed for the MRA-TC architecture. By applying spatial compensation in P and B pictures (MRA-SC), the quality loss is reduced to less than 0.1 dB.

MRA-Hybrid improves quality somewhat by applying temporal compensation.

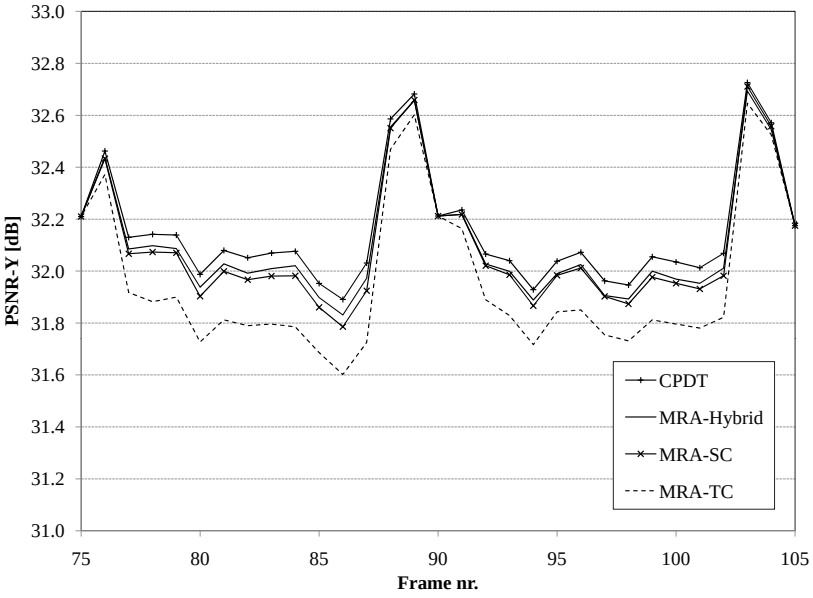


Fig. 19. Sequence of transcoded frames (Stockholm, 720p, $QP_1 = 32$, $\Delta QP = 3$).

For sequences containing high motion, such as Stefan, the impact of compensation becomes even more visible, as displayed in Fig. 20. MRA-SC limits the quality loss and drift, which becomes clearer at the end of the GOP. Also, the impact of temporal drift is more pronounced in high-motion sequences; this is tackled by the MRA-Hybrid architecture.

As was remarked in Sect. 2.3.5, non-linear operations will still result in drift in the output bitstream, however. This becomes apparent when long GOP sizes are used. As an indication of the drift, Fig. 21 shows an 'extreme' comparison between the MRA-TC, MRA-SC and MRA-Hybrid architectures for an IBBP GOP structure (298 frames) without intra refresh (only the first picture is intra-coded, as in the experiment in Sect. 2.3). A large increase in QP ($\Delta QP = 6$) was selected to clearly demonstrate the drift effect. This experiment shows the improvement in quality obtained by spatial compensation, which is able to significantly restrain the drift (as can be seen, e.g., at the

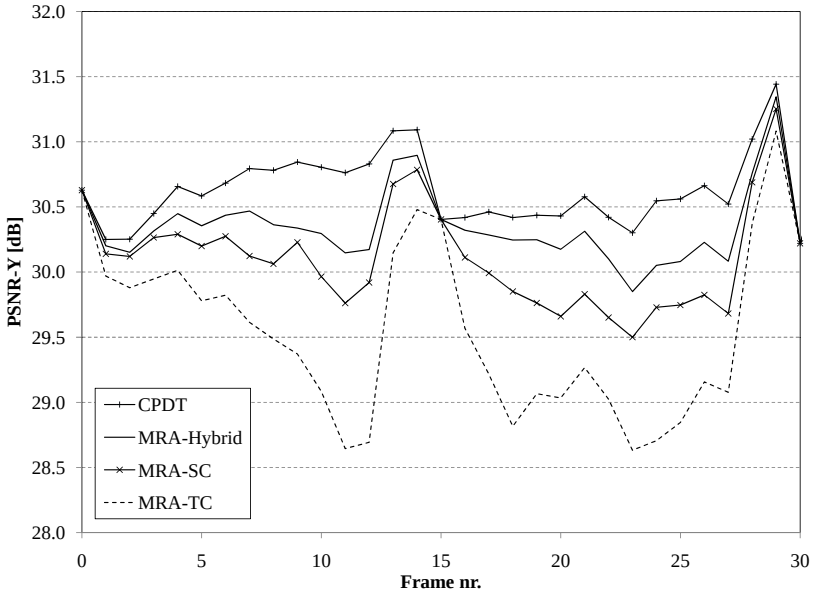


Fig. 20. Sequence of transcoded frames (Stefan, CIF, $QP_1 = 32$, $\Delta QP = 3$).

end of the sequence). Nonetheless, for optimum quality, temporal compensation on top of spatial compensation remains as equally important. By adding the temporal compensation, a more gradual decrease in quality is noticed. The temporal compensation comes at the expense of a smaller bit rate reduction (a 40% reduction was noticed for the MRA-Hybrid architecture, as opposed to a 56% bit rate reduction for MRA-SC), and of higher computational complexity. Equally importantly, this figure demonstrates that the drift effect will become severe when long GOP structures are used. This limits the applicability of compensation architectures for larger GOP sizes when coarse requantization is desired (i.e., a large increase in QP). For such large GOP sizes, a combination of the CPDT and MRA-SC or MRA-Hybrid approaches could be used, e.g., by applying CPDT for both I and P pictures, while keeping complexity low by using single-loop or open-loop transcoding for B pictures.

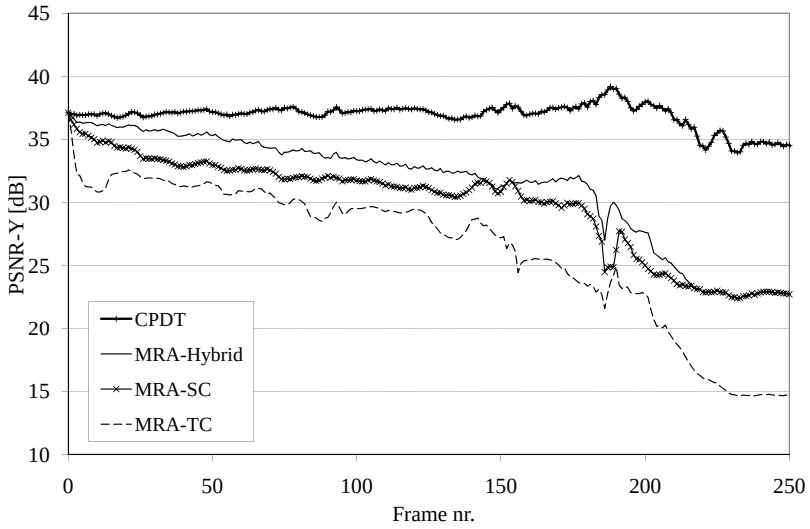


Fig. 21. Sequence of transcoded frames (Foreman, CIF, $QP_1 = 22$, $\Delta QP = 6$).

4.2 Timing results

In Table 3, timing results are shown for the different transcoder architectures. These results are obtained from our non-optimized transcoder software, and serve as an indication of the relative complexity of the techniques and architectures. The processing speed is shown in frames per second.

It can be seen from the difference between MRA-TC and MRA-Hybrid that adding spatial compensation has only a minor impact on processing speed. Traditional temporal compensation, however, has a large effect for H.264/AVC requantization transcoding, for the reasons mentioned in Sect. 3.5. The MRA-SC architecture is able to obtain transcoding speeds up to 80% of the open-loop transcoder, with significantly improved visual and rate-distortion results (as shown in Sect. 4.1).

Table 3

Processing speed for transcoding architectures [fps]

	Foreman (QCIF)	Stefan (CIF)	Crew (4CIF)	Stockholm (720p)
CPDT	21.51	4.76	1.44	0.48
MRA-Hybrid	34.86	7.53	2.33	0.78
MRA-TC	35.61	7.74	2.52	0.78
MRA-SC	277.50	32.17	10.07	6.68
OL	410.79	40.14	16.43	8.87

5 Conclusions

When compared to previous video coding standards, such as MPEG-1/2, a number of changes are required in order to make requantization transcoding practically usable. Spatial prediction introduces a new challenge in H.264/-AVC, and causes significant drift and visual artifacts when not taken care of properly. Although previous publications reported single-loop solutions not to be practically viable, we showed that a hybrid single-loop architecture with spatial and temporal compensation results in rate-distortion performance within 1 to 2 dB of the decoder-encoder cascade, and even more importantly, the removal of disturbing visual artifacts in the transcoded pictures. When compared to previous solutions, such as in [15], this is made possible due to the addition of spatial compensation. The fast architecture using only spatial compensation (MRA-SC) results in highly improved rate-distortion curves over traditional single-loop architectures, at a speed up to 80% of the open-

loop transcoder. Spatial compensation enables the possibility of a new class of low-complexity transcoders that are able to restrain the quality loss due to requantization. Temporal compensation can additionally be applied to further improve quality, at the cost of increased computational complexity.

Acknowledgment

The research activities that have been described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT-Flanders), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

References

[1] A. Eleftheriadis and D. Anastassiou, "Constrained and general dynamic rate shaping of compressed digital video," in *IEEE Int. Conf. Image Process. (ICIP)*, October 1995.

[2] A. Eleftheriadis and P. Batra, "Dynamic rate shaping of compressed digital video," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 297–314, February 2006.

[3] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG bitstreams," *Signal Process.: Image Communication*, vol. 8, no. 6, pp. 481–500, September 1996.

[4] H. Sun, W. Kwok, and J. W. Zdepski, "Architectures for MPEG compressed bitstream scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp. 191–199, April 1996.

[5] Y. Nakajima, H. Hori, and T. Kanoh, "Rate conversion of MPEG coded video by re-quantization process," in *IEEE Int. Conf. Image Process. (ICIP)*, October 1995.

[6] P. Assunção and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 953–967, December 1998.

[7] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.

[8] D. Lefol and D. Bull, "Mode refinement algorithm for H.264 inter frame requantization," in *IEEE Int. Conf. Image Process. (ICIP)*, October 2006.

[9] D. Lefol, D. Bull, and N. Canagarajah, "Mode refinement algorithm for H.264 intra frame requantization," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2006.

[10] P. Zhang, Y. Lu, Q. Huang, and W. Gao, "Mode mapping method for H.264/AVC spatial downscaling transcoding," in *IEEE Int. Conf. Image Process. (ICIP)*, October 2004.

[11] H. Shen, X. Sun, F. Wu, H. Li, and S. Li, "A fast downsizing video transcoder for H.264/AVC with rate-distortion optimal mode decision," in *IEEE Int. Conf. Multimedia Expo (ICME)*, July 2006.

[12] C.-H. Li, C.-N. Wang, and T. Chiang, "A multiple-window video embedding transcoder based on H.264/AVC standard," *EURASIP Journal Advances Signal Process.*, 2007.

[13] J. Bialkowski, M. Barkowsky, F. Leschka, and A. Kaup, "Low-complexity transcoding of inter coded video frames from H.264 to H.263," in *IEEE Int. Conf. Image Process. (ICIP)*, October 2006.

[14] G. Fernandez-Escribano, J. Bialkowski, J. Gamez, H. Kalva, P. Cuenca, L. Orozco-Barbosa, and A. Kaup, "Low-complexity heterogeneous video transcoding using data mining," *IEEE Trans. Multimedia*, vol. 10, no. 2, pp. 286–299, February 2008.

[15] D. Lefol, D. Bull, and N. Canagarajah, "Performance evaluation of transcoding algorithms for H.264," *IEEE Trans, Consum. Electron.*, vol. 52, no. 1, pp. 215–222, February 2006.

[16] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, July 2003.

[17] O. Werner, "Requantization for transcoding of MPEG-2 intraframes," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 179–191, February 1999.

[18] B. Shen, "Perfect requantization for video transcoding," *Multimedia Tools Applicat.*, vol. 35, no. 2, pp. 163–173, November 2007.

[19] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18–29, March 2003.

[20] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video transcoding: an overview of various techniques and research issues," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793–804, October 2005.

[21] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70–84, February 1999.

[22] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[23] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, July 2003.

[24] G. Triantafyllidis, D. Tzovaras, and M. Strintzis, "Blocking artifact detection and reduction in compressed data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 877–890, October 2002.

[25] B. Shen, "Efficient deblocking and optimal quantizer selection for video transcoding," in *IEEE Int. Conf. Image Process. (ICIP)*, September 2003.

[26] H. Sun, X. Chen, and T. Chiang, *Digital video transcoding for transmission and storage.* CRC Press, 2005.

[27] J. De Cock, S. Notebaert, and R. Van de Walle, "A novel hybrid requantization transcoding scheme for H.264/AVC," in *Proc. Int. Symp. Signal Process. & its Appl.*, 2007.

[28] C. Chen, P.-H. Wu, and H. Chen, "Transform-domain intra prediction for H.264," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2005.

[29] X. Sun and P. Tao, "Rapid algorithms for MPEG-2 to H.264 transcoding," in *Pacific Rim Conf. Multimedia (PCM)*, November 2005.

[30] S. Notebaert, J. De Cock, P. Lambert, D. De Schrijver, and R. Van de Walle, "Requantization transcoding of H.264/AVC bitstreams for intra 4x4 prediction modes," in *Proc. Pacific-rim Conf. on Multimedia*, 2006.

[31] J. De Cock, S. Notebaert, P. Lambert, D. De Schrijver, and R. Van de Walle, "Requantization transcoding in pixel and frequency domain for intra 16x16 in H.264/AVC," in *Proc. Advanced Concepts for Intelligent Vision Systems*, 2006.

[32] T. Wedi and H. G. Musmann, "Motion- and aliasing-compensated prediction for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 577–586, July 2003.

[33] O. Werner, "Drift analysis and drift reduction for multiresolution hybrid video coding," *Signal Process.: Image Communication*, vol. 8, no. 5, pp. 387–409, July 1996.

[34] S. Notebaert, J. De Cock, and R. Van de Walle, "Improved H.264/AVC requantization transcoding using low-complexity interpolation filters for 1/4-pixel motion compensation," in *IEEE Symp. Comput. Intell. Image Signal Process.*, 2007.

[35] T. Tan, G. Sullivan, and T. Wedi, *Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 1*, Video Coding Experts Group, Doc. VCEG-AE10, Marrakech, Morocco, January 2007.